Extracting a Sparsely-Located Named Entity from Online HTML Medical Articles Using Support Vector Machine

Jie Zou*, Daniel Le, George R. Thoma
Lister Hill National Center for Biomedical Communications, National Library of Medicine
8600 Rockville Pike, Bethesda, MD 20894

ABSTRACT

We describe a statistical machine learning method for extracting databank accession numbers (DANs) from online medical journal articles. Because the DANs are sparsely-located in the articles, we take a hierarchical approach. The HTML journal articles are first segmented into zones according to text and geometric features. The zones are then classified as DAN zones or other zones by an SVM classifier. A set of heuristic rules are applied on the candidate DAN zones to extract DANs according to their edit distances to the DAN formats. An evaluation shows that the proposed method can achieve a very high recall rate (above 99%) and a significantly better precision rate compared to extraction through brute force regular expression matching.

Keywords: Information Extraction, Support Vector Machine, Databank Accession Number, Named Entity Recognition.

1. INTRODUCTION

The automatic extraction of bibliographic data from medical journals is key to the affordable creation of citations for MEDLINE®, the flagship database of the U.S. National Library of Medicine (NLM) containing over 14 million citations, and searched over 3 millions times a day worldwide.

A variety of rule-based and machine learning algorithms are employed in the extraction of this data which includes the article title, author names, affiliations, abstract, and more recently, *databank accession numbers* (DANs). This has come about since submission of sequence information to nucleotide sequence databases prior to publication has become a standard practice. A unique accession number is assigned by the database (e.g., GenBank[19], Protein Data Bank[18], etc.) which permanently identifies the sequence submitted, and this number appears in articles either on the first page, or as required by individual journal procedures. This procedure ensures availability and distribution of new sequence data in a timely fashion.

Currently, six kinds of databank accession numbers are required to be extracted. The types and their format information are listed in Table 1. We also list the corresponding *regular expressions*, which precisely define the formats.

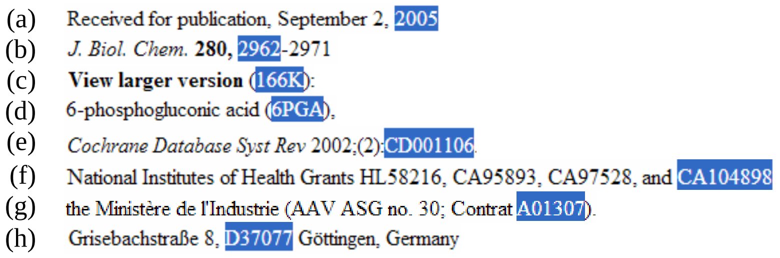
Since databank accession numbers have well-defined formats, a straightforward method for DAN extraction is to search text according to their formats, which can easily be implemented through *Regular Expression Matching*. However, many other entities in the journal articles can have the same formats as DANs, such as 4-digit years and page numbers. Typical examples are shown in Figure 1. With straightforward Regular Expression Matching, these other entities can generate a large number of false positives. Errors also arise when authors do not precisely follow the required DAN formats, or make typographical errors. Several examples of these are shown in Figure 2, in which Regular Expression Matching will create false negatives.

An experiment, described in Section 4.2, shows that most authors strictly follow the required formats, which allows simple brute force Regular Expression Matching to achieve high recall rates. However, in a typical journal article, due to the large number of false positives, the precision rate can be as low as 3.9%. Regular Expression Matching, therefore, is insufficient for DAN extraction. Further processing is required to meet the goal of significantly increasing the precision rate without greatly sacrificing the recall rate.

*jzou@mail.nlm.nih.gov; phone 1 301 496-7086; fax 1 301 402-0341;

Table 1: Databank accession number types and their formats. Examples are given after “Ex:”.

Type	Formatting notes	Format and examples
Protein Data Bank (PDB)[18]	One digit number followed by three digits or uppercase letters	[0-9]([0-9]|[A-Z]){3} Ex: 2B38, 2C2P, 2ETR
GenBank[19]	Three kinds of formats: 1. 3 uppercase letters followed by 5 digits 2. 2 uppercase letters followed by 6 digits 3. 1 uppercase letters followed by 5 digits	 [A-Z]{3}[0-9]{5} Ex: AAP51207 [A-Z]{2}[0-9]{6} Ex: AY572787 [A-Z][0-9]{5} Ex: Z54326
Gene Expression Omnibus (GEO)[20]	Prefix, which must be one of GDS, GSE, GPL or GSM, followed by one or more digits	(GDS|GSE|GPL|GSM)[0-9]+ Ex: GSM40956, GSE3181
Reference Sequence (RefSeq)[21]	Prefix, which must be one of AC, AP, NC, NG, NM, NP, NR, NT, NW, NZ, XM, XP, XR, YP or ZP, followed by ‘_’, and then 6 or 9 digits.	(AC|AP|NC|NG|NM|NP|NR|NT|NW|NZ|XM|XP|XR|YP|ZP)_([0-9]{6}|[0-9]{9}) Ex: NC_004459, XM_236792
ClinicalTrials[22]	Prefix, NCT, followed by 8 digits	NCT[0-9]{8} Ex: NCT00112255
International Standard Randomized Controlled Trial Number (ISRCTN)[23]	Prefix, ISRCTN, followed by 8 digits	ISRCTN[0-9]{8} Ex: ISRCTN31571714



(a) Received for publication, September 2, 2005

(b) *J. Biol. Chem.* **280,** 2962-2971

(c) **View larger version** (166K):

(d) 6-phosphogluconic acid (6PGA),

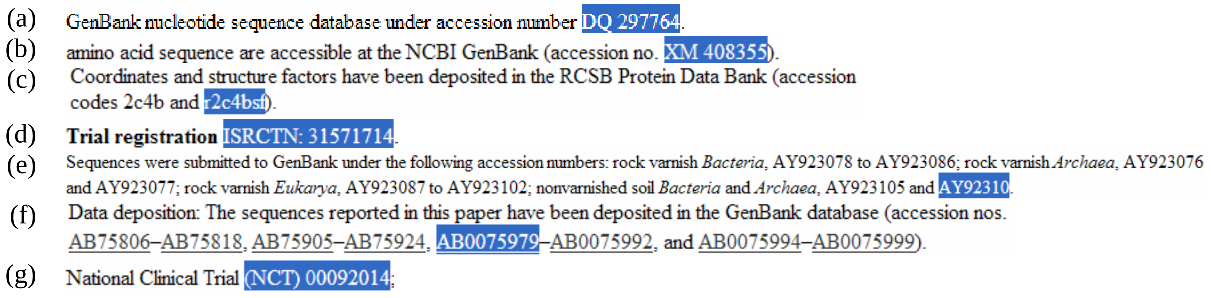
(e) *Cochrane Database Syst Rev* 2002;(2):CD001106.

(f) National Institutes of Health Grants HL58216, CA95893, CA97528, and CA104898

(g) the Ministère de l'Industrie (AAV ASG no. 30; Contrat A01307).

(h) Grisebachstraße 8, D37077 Göttingen, Germany

Figure 1: Examples of other entities mimicking legitimate DAN formats. Mistaken for a legitimate PDB number are (a) 4-digit year; (b) 4-digit page number; (c) file size description; (d) Chemical term. Mistaken for a legitimate GenBank number are: (e) page number; (f) grant number; (g) foreign contract number; (h) foreign zip code.



(a) GenBank nucleotide sequence database under accession number DQ 297764.

(b) amino acid sequence are accessible at the NCBI GenBank (accession no. XM 408355).

(c) Coordinates and structure factors have been deposited in the RCSB Protein Data Bank (accession codes 2c4b and r2c4bsf).

(d) **Trial registration** ISRCTN: 31571714.

(e) Sequences were submitted to GenBank under the following accession numbers: rock varnish *Bacteria*, AY923078 to AY923086; rock varnish *Archaea*, AY923076 and AY923077; rock varnish *Eukarya*, AY923087 to AY923102; nonvarnished soil *Bacteria* and *Archaea*, AY923105 and AY92310.

(f) Data deposition: The sequences reported in this paper have been deposited in the GenBank database (accession nos. AB75806–AB75818, AB75905–AB75924, AB0075979–AB0075992, and AB0075994–AB0075999).

(g) National Clinical Trial (NCT) 00092014;

Figure 2: Examples of poorly-formatted databank accession numbers. (a) extra space between prefix and number; (b) “_” character replaced by space; (c) non-fully compatible format; (d) extra colon and space; (e) typo, the correct GenBank number is most likely AY923106; (f) extra “0”s; (g) extra parentheses and space.

In addition to format, the surrounding text offers important features for DAN extraction, significantly narrowing the candidates. We therefore propose a detection procedure with the following three steps:

1. Segment the HTML article into logical zones (blocks) using text and geometric features.
2. Classify each zone as a DAN zone based on the text in the zone.
3. Extract DANs from the candidate zones.

There are two advantages of taking this coarse-to-fine approach. One is that other entities that mimic legitimate DAN formats, such as those shown in Figure 1, can be safely ignored, and therefore significantly increase the precision rate. The other is that due to significantly reduced candidates, sophisticated methods can be designed to extract poorly formatted DANs, such as those shown in Figure 2, and therefore possibly increase the recall rate.

In this research, we primarily concentrate on the second step, i.e., identifying candidate DAN zones. Our current implementation for the third step is a set of simple heuristic rules. More sophisticated methods may still be computationally feasible since only a few candidate zones are required to go through this step. We review some related works in Section 2. The details of the algorithm are presented in Section 3. We present an evaluation of the proposed method in Section 4. Summary and conclusions constitute Section 5.

2. RELATED WORK

The extraction of databank accession numbers falls in the general category of *named-entity-recognition (NER)*, which typically involves the identification of locations, person names, organizations, dates, times, monetary amounts, etc., and has been well researched. In the newswire domain, the best NER algorithm can now achieve 0.95 F-score, which is considered close to human performance[1, 11]. Biomedical NER, used to identify technical terms in the biology domain (e.g. gene, protein, etc.), is of increasing interest[7]. Compared to the newswire domain, however, biomedical NER is more challenging. Several machine learning approaches have been proposed for this domain, including Support Vector Machine[8] and Conditional Random Field[14], as well as combinations of several methods to further improve performance[15].

There are two important differences between databank accession numbers and typical named entities: (1) the DANs have well-defined formats; and (2) they are sparsely located in the text. Most NER algorithms model and analyze text at the sentence level. Because DANs are sparsely located, and since most zones in an article are irrelevant, it is more efficient to take a coarse-to-fine approach and conduct a zone level analysis, thereby filtering out most irrelevant zones. Then, for the remaining few candidate DAN zones, existing NER methods can be adopted to analyze sentences and extract DANs. However, because of the well-specified formats of the DANs, we choose to use a set of simple heuristic rules, an approach that is more computationally efficient.

The detection of candidate DAN zones may be viewed as a text categorization problem. Machine learning-based text categorization has also been intensively studied for more than a decade. Among the existing methods, Support Vector Machine[3] and boosting-based classifier committees[12] are considered the best ones. We chose Support Vector Machine for our DAN zone classification.

3. METHOD

Besides DAN extraction, HTML segmentation is also very useful for many other information retrieval tasks, and has been discussed in our previous works[16, 17]. In this section, we describe only the second and third steps of our DAN extraction method.

3.1 DAN zone classification using Support Vector Machine

DAN zone classification is a text categorization problem, i.e., categorizing zones into *DAN zones* (the zones containing databank accession numbers) and *other zones* (the zones not containing databank accession numbers).

The first step in classifying a given zone is to extract useful features to represent it, such as word frequency counts. An important question is how to choose the dictionary, i.e., the set of words to be counted. After removing stop words and rarely-appeared (less than 10 counts) words, 23,202 distinct words are collected from our training articles. It is well-known in text categorization that the high dimensionality of the word space, i.e., the large size of the dictionary, may be problematic due to "the curse of dimensionality." To avoid this, a dimension reduction method is employed to select an optimal word dictionary.

In a survey of text categorization by Sebastiani[13], the GSS measure[5] is recognized as one of the best methods for feature dimension reduction ("GSS" named after the three authors of the referenced papers). Originally, the GSS measure is defined for each class label. In our DAN zone classification, GSS measures of word t_k for DAN zone label c_0 and "other zone" label c_1, are defined as:

$$GSS(t_k, c_0) = P(t_k, c_0)P(\bar{t}_k, c_1) - P(t_k, c_1)P(\bar{t}_k, c_0)$$

$$GSS(t_k, c_1) = P(t_k, c_1)P(\bar{t}_k, c_0) - P(t_k, c_0)P(\bar{t}_k, c_1)$$

where, $P(\bar{t}_k, c_i)$ indicates the probability that, given a random zone, word t_k does not occur in the zone and that the zone belongs to category c_i. In our two-class classification, we define a joint GSS measure for each word t_k to be:

$$GSS(t_k) = \left| P(t_k, c_1)P(\bar{t}_k, c_0) - P(t_k, c_0)P(\bar{t}_k, c_1) \right| .$$

The GSS measure reflects the intuition that the best words are the ones distributed most differently in the DAN and other zones. $P(t_k, c_i)$ and $P(\bar{t}_k, c_i)$ can be estimated by counting occurrences in the training samples.

The 23,202 words in our training samples can then be sorted according to their GSS measures. A higher value for this measure generally indicates better discriminant ability. Table 2 shows the 20 words with the highest GSS measures. It is of interest to find out how many words, i.e., the dictionary size, are required to achieve good performance in our DAN zone classification. An empirical study is described in Section 4.5 to answer this question.

Table 2: The top 20 words with highest GSS measures.

accession	deposited	genbank	data	bank	sequence	nucleotide	coordinates	sequences	abstract
text	database	numbers	protein	code	number	reported	crossref	atomic	paper

Once the word dictionary is selected, the occurrences of these words in the zone can be counted. These counts form a word-frequency feature vector, denoted as $\mathbf{f_i} = \{f(t_1, d_i), \cdots, f(t_k, d_i), \cdots, f(t_n, d_i)\}$, where t_k is the k^th word in the dictionary, n is the dictionary size, d_i is a zone, and $f(t_k, d_i)$ is the number of occurrences of word t_k in zone d_i. In order to make zones of different length comparable, the word-frequency feature vector is normalized by the total number of words in the zone. These normalized feature vectors are the representations of the zones, and are used to train the SVM classifier which then predicts the labels of the test zones.

The DAN zone classification problem is very unbalanced. In a typical journal article, there are significantly more "other zones" than DAN zones. As a consequence, we have many more training samples for "other zones" than for DAN zones. It is known that SVM classifiers are sensitive to unbalanced training samples, and are biased toward the class label with more training samples. An empirical study is presented in Section 4.4 to find the best combination of DAN and "other zone" training samples.

3.2 Heuristic rules for DAN detection

Once zones are labeled as DAN zones, we use a set of heuristic rules to detect the databank accession numbers. Following the commonly-used edit distance in approximate string matching[10], we define the edit distance of words in the zone to DAN formats as the cost of deleting, inserting and replacing characters to match specific formats. The cost for one operation (deleting, inserting or replacing) is 1. For example, the edit distances of "E12345" to PDB and GenBank are 2 and 0, respectively.

The heuristic rules in our current algorithm are:

- If the format of a word matches the PDB format, it is labeled as a PDB number. If the word is within one edit distance of GenBank, GEO or RefSeq format, it is labeled as one of these. Similarly, if the word is within three edit distances of ClinicalTrials or ISRCTN formats, it is labeled as ClinicalTrials or ISRCTN, respectively. For example, in Figure 2(e), the word "AY92310" is one edit distance away from the format of the type 2 GenBank. According to this rule, this word is extracted and labeled as a GenBank.

- For two adjacent words, if the first one satisfies a DAN prefix format and the second one satisfies the suffix format of the same DAN, merge the two words and label the combination as a DAN. For example, in Figure 2(g), two adjacent words are "NCT" and "00092014" (In our implementation, any non-letter and non-digit characters, except for underscore character, "_", are trimmed). "NCT" is the prefix of the ClinicalTrials, and "00092014" satisfies the suffix format of ClinicalTrials, therefore, these two words are merged and labeled as a DAN of ClinicalTrials.

4. EXPERIMENTAL EVALUATION

In this section, we begin with describing the data used for evaluation and presenting the performance with straightforward Regular Expression Matching. After briefly discussing the SVM classifier we used for databank zone detection, we present empirical analysis of two aspects of the SVM classifier, specifically for DAN zone classification. Finally, we present the overall precision and recall rates yielded by the complete DAN extraction algorithm.

4.1 Experimental data

We searched through MEDLINE 2006 database (citations of articles indexed in 2006) to collect 1617 articles containing DANs. 1000 articles were randomly selected as training samples, and the remaining 617 articles as test samples. All articles were segmented into zones by our HTML journal article segmentation algorithm[16, 17]. Through simple string matching, the zones containing DANs were extracted and labeled as DAN zones. The remaining ones were labeled as "other zones". Table 3 summarizes the statistics of the experimental data, used to evaluate the two stages of our DAN extraction algorithm.

Table 3: Experimental data statistics

	Articles	DANs	DAN Zones	Other Zones
Training	1000	3076	1491	66,458
Testing	617	1468	877	41,419

4.2 Brute force extraction with Regular Expression Matching

Since DANs have specific formats, a straightforward method for detecting DANs is to use Regular Expression Matching. We conducted an experiment on the 617 test articles to evaluate the performance of this brute force approach. Out of a total of 1486, 18 DANs are missed (false negatives), but there are 36,565 false positives. This indicates that most authors are indeed very careful about entering DANs, and due to the rigid formats of DANs, the recall rate is high, 98.8%. However, because there are plenty of other entities having the same format as DANs, the precision rate is very low, 3.9%.

4.3 SVM classifier

We use LibSVM[2], an SVM library developed at National Taiwan University, to implement our DAN zone classification. We adopted Radial Basis Function (RBF) as the kernel function, and the two parameters, C (penalty parameter of the errors) and γ (RBF parameter), were selected through exhaustive grid-search using cross-validation on training samples. The features for representing zone texts are the word frequency counts. Because we choose to use only 400 words, the DAN zone classification is a fast process, taking about a few hundreds of milliseconds to process a typical article on a 3.40GHz PC equipped with 1GB RAM.

4.4 Effect of unbalanced training samples in SVM DAN zone classification

It is a known problem for SVM classifiers that unbalanced training data can seriously degrade classification performance. In DAN zone classification, there are significantly more "other zones" than DAN zones. We conducted an experiment to test how unbalanced training samples affect the classification. The 100 words with the highest GSS measures are used as the dictionary for word frequency counting. The evaluation is on a total of 1877 test zones, 877 of them are the available test DAN zones, and the other 1000 randomly selected from the 41,419 test "other zones". There is a total of 1491 training DAN zones, all used as training samples in our experiments. In addition, we include 372, 745, 1491, 2982 and 5964 randomly selected "other zones" into the training set. These are chosen because they are respectively ¼, ½, 1, 2, 4 times the number of training DAN zones, i.e., 1491. Table 4 shows the false positive (other zones mislabeled as DAN zones), false negative (DAN zones mislabeled as other zones), and average error rates.

Table 4: Results of varying the number of training samples.

	False Positive	False Negative	Average Error
1491 DAN and 372 other zones	19.9%	5.1%	12.5%
1491 DAN and 745 other zones	7.1%	8.3%	7.7%
1491 DAN and 1491 other zones	3.7%	10.4%	7.1%
1491 DAN and 2982 other zones	1.8%	13.0%	7.4%
1491 DAN and 5964 other zones	1.4%	15.2%	8.3%

This experiment clearly demonstrates that the SVM classifier is biased toward the class label with more training samples, and that classification performance is not always improved by adding more training samples. The balance of training samples also plays a very important role. In our DAN extraction problem, false negative errors (under detection) are considered much more serious than false positive (over detection) errors. Therefore, 1491 DAN and 745 "other zones" are chosen to train the SVM classifier.

4.5 Effect of dictionary size

Although pointed out by Sebastiani[13] that feature reduction is usually required in machine learning-based text categorization, several researchers have also shown that SVM classifiers are capable of effectively processing feature vectors of more than 10,000 dimensions[4, 6, 9]. The feature dimension, i.e., the word dictionary size, however affects not only classification accuracy, but also the computation time, which is another critical consideration in our operational system. Therefore, it is of interest to find out how dictionary size affects the performance of our DAN zone classification. Table 5 and Figure 3 show the results of varying the dictionary size from 50 to 6400.

Table 5: Results of varying dictionary size.

Dictionary Size	False Positive	False Negative	Average Error
50	4.7%	10.3%	7.5%
100	7.1%	8.3%	7.7%
200	9.8%	7.1%	8.5%
400	10.7%	6.4%	8.6%
800	12.9%	6.6%	9.8%
1600	12.7%	7.2%	10.0%
3200	11.5%	7.0%	9.3%
6400	11.6%	7.4%	9.5%

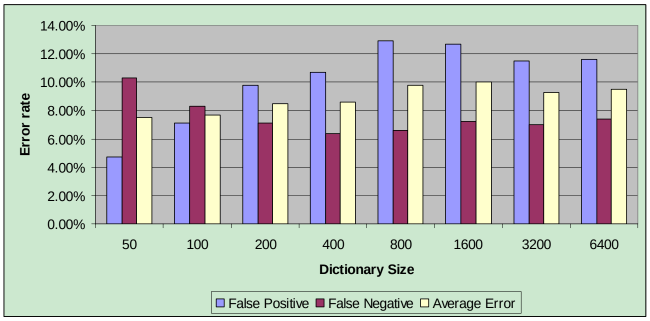
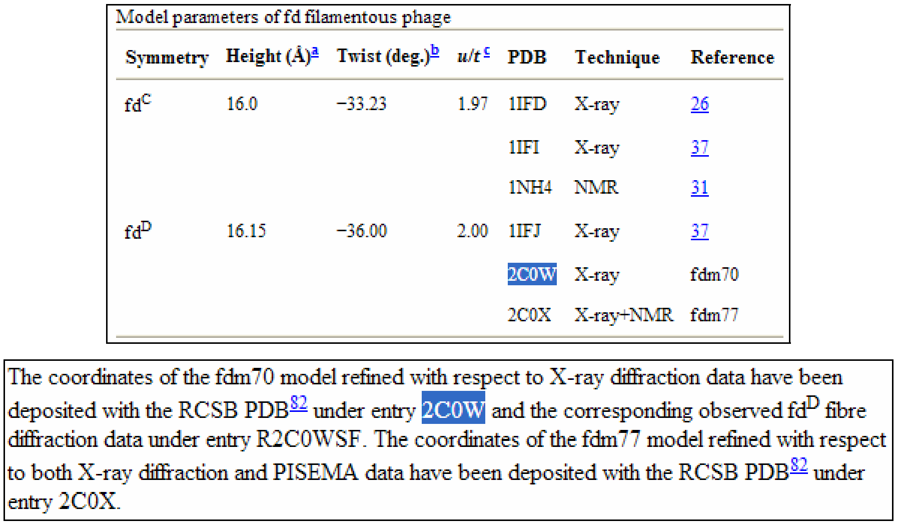


Figure 3: Results of varying dictionary size in bar chart.

SVM, as found by other researchers, is indeed robust with respect to high dimensional feature vectors. The performance drops only slightly even with a dictionary size 128 times larger. Again, we care more about false negatives, and therefore, we select a dictionary size of 400. This relatively low dimensionality also renders the SVM classifier computationally efficient.

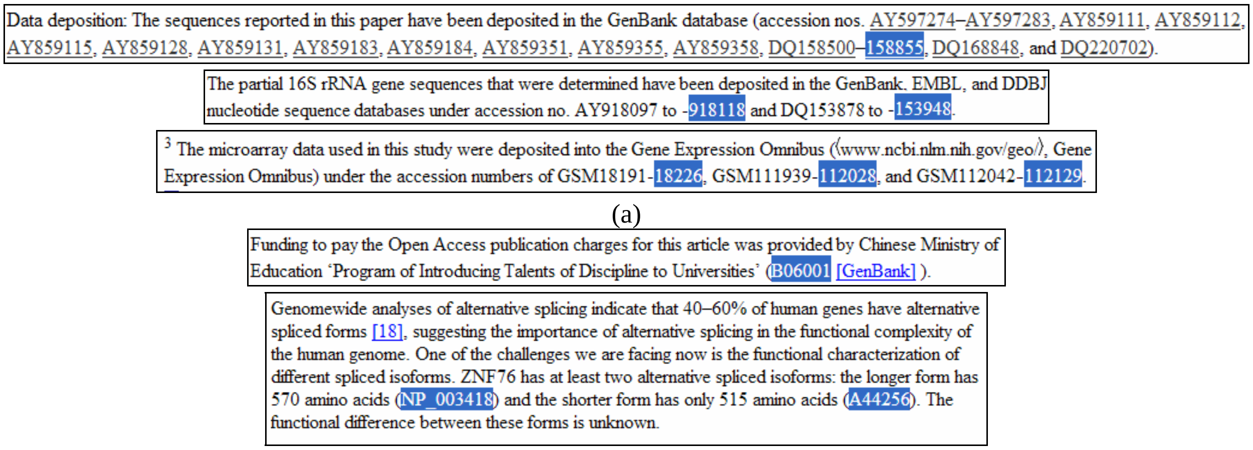


Model parameters of fd filamentous phage

Symmetry	Height (Å)[a]	Twist (deg.)[b]	*u*/*t*[c]	PDB	Technique	Reference
fd^{C}	16.0	−33.23	1.97	1IFD	X-ray	26
				1IFI	X-ray	37
				1NH4	NMR	31
fd^{D}	16.15	−36.00	2.00	1IFJ	X-ray	37
				2C0W	X-ray	fdm70
				2C0X	X-ray+NMR	fdm77

The coordinates of the fdm70 model refined with respect to X-ray diffraction data have been deposited with the RCSB PDB[82] under entry 2C0W and the corresponding observed fd^{D} fibre diffraction data under entry R2C0WSF. The coordinates of the fdm77 model refined with respect to both X-ray diffraction and PISEMA data have been deposited with the RCSB PDB[82] under entry 2C0X.

Figure 4: Databank accession numbers may appear in several places in an article. In the top, the PDB number 2C0W is listed in a table, which is classified as an "other zone" due to the lack of contextual information. In the bottom, the same DAN is mentioned again in a paragraph, which is classified as a DAN zone, and is therefore correctly detected.



Data deposition: The sequences reported in this paper have been deposited in the GenBank database (accession nos. AY597274–AY597283, AY859111, AY859112, AY859115, AY859128, AY859131, AY859183, AY859184, AY859351, AY859355, AY859358, DQ158500–158855, DQ168848, and DQ220702).

The partial 16S rRNA gene sequences that were determined have been deposited in the GenBank, EMBL, and DDBJ nucleotide sequence databases under accession no. AY918097 to -918118 and DQ153878 to -153948.

[3] The microarray data used in this study were deposited into the Gene Expression Omnibus (〈www.ncbi.nlm.nih.gov/geo/〉, Gene Expression Omnibus) under the accession numbers of GSM18191-18226, GSM111939-112028, and GSM112042-112129.

(a)

Funding to pay the Open Access publication charges for this article was provided by Chinese Ministry of Education 'Program of Introducing Talents of Discipline to Universities' (B06001 [GenBank]).

Genomewide analyses of alternative splicing indicate that 40–60% of human genes have alternative spliced forms [18], suggesting the importance of alternative splicing in the functional complexity of the human genome. One of the challenges we are facing now is the functional characterization of different spliced isoforms. ZNF76 has at least two alternative spliced isoforms: the longer form has 570 amino acids (NP_003418) and the shorter form has only 515 amino acids (A44256). The functional difference between these forms is unknown.

(b)

Figure 5: False negative errors. (a) Abbreviated databank accession numbers; (b) Missed by the SVM classifier.

4.6 Overall DAN extraction performance

After the zone classification, DANs are extracted by applying the set of heuristic rules defined in Section 3.2 to the candidate DAN zones. Worth mentioning here is that the DANs are sometimes mentioned in the article in several places. Figure 4 shows an example where the same DAN, "2C0W", is mentioned twice in the article, the top zone being classified as an "other zone". However, this is not a catastrophe, since the DAN may be extracted from the bottom zone. This is not a rare case. The 6.4% false negative rate in our DAN zone classification, therefore, does not mean that 6.4% DANs will be missed by the algorithm.

Actually, only 9 DANs, out of a total of 1486, are missed, achieving 99.4% recall rate. All 9 false negatives are highlighted in Figure 5. The 6 DANs in Figure 5(a) are missed because of abbreviated format. They could be recovered by adding more rules in Step 3. The three DANs missed in Figure 5(b) is due to misclassification by the SVM classifier; their recovery is more difficult, particularly because few clues exist in the surrounding text to indicate that they are DANs.

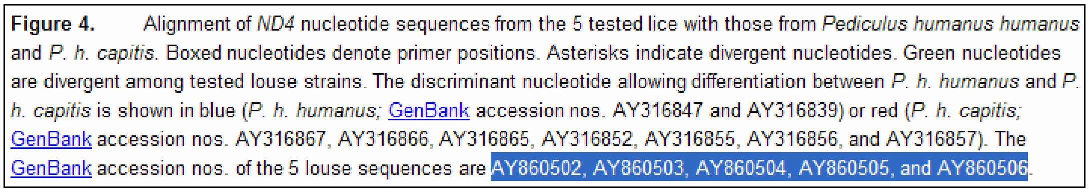


Figure 4. Alignment of *ND4* nucleotide sequences from the 5 tested lice with those from *Pediculus humanus humanus* and *P. h. capitis*. Boxed nucleotides denote primer positions. Asterisks indicate divergent nucleotides. Green nucleotides are divergent among tested louse strains. The discriminant nucleotide allowing differentiation between *P. h. humanus* and *P. h. capitis* is shown in blue (*P. h. humanus;* GenBank accession nos. AY316847 and AY316839) or red (*P. h. capitis;* GenBank accession nos. AY316867, AY316866, AY316865, AY316852, AY316855, AY316856, and AY316857). The GenBank accession nos. of the 5 louse sequences are AY860502, AY860503, AY860504, AY860505, and AY860506.

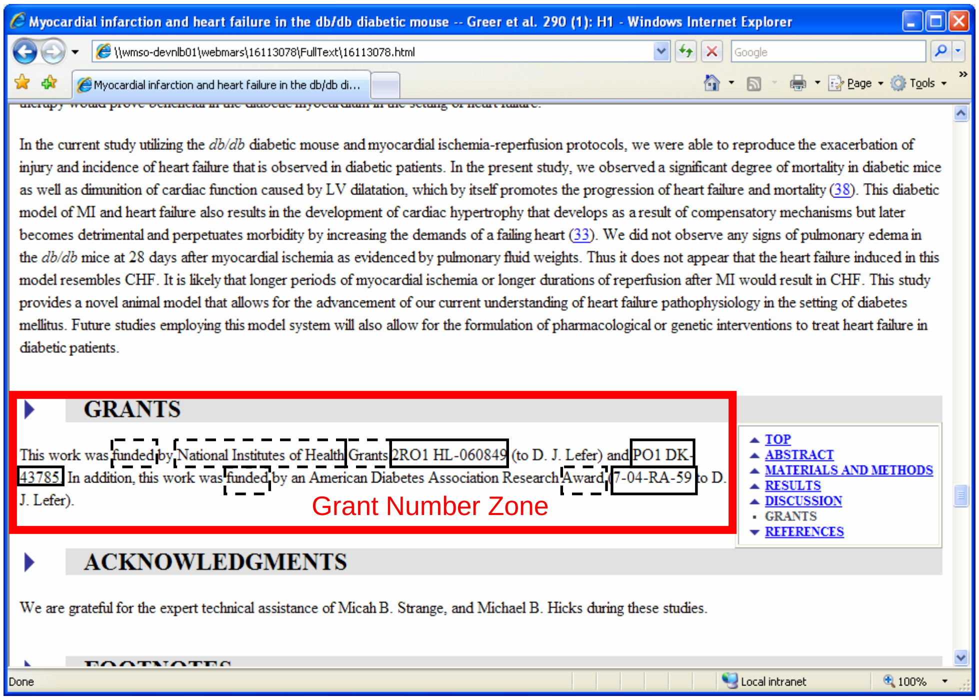
Figure 6: MEDLINE collects only "new" DANs (the highlighted 5). Because we use MEDLINE data as ground truth, 9 false positives are counted for this article.

The false positives are significantly reduced: 2920 compared to 36,565 by Regular Expression Matching, achieving 33.5% precision rate, significantly higher than the 3.9% from the brute force approach. Even though the precision rate of 33.5% appears low, this is misleading, since most false positives are indeed DANs. By library policy, MEDLINE collects only the "new" DANs. For example, in Figure 6, only the 5 highlighted DANs are collected by MEDLINE for this article. The other 9 are abandoned. Because we use MEDLINE data as our ground truth, these 9 DANs are counted as false positives. In actual operation, these "old" false positive DANs can be easily filtered out by checking whether they have already been associated with other articles, leading to a much higher precision rate in actual operation. We intend to provide an accurate estimate of actual precision in our future work.

5. CONCLUSION

We describe a statistical machine learning approach for databank accession number extraction from online (HTML) medical journal articles. We find that a brute force approach using Regular Expression Matching is insufficient for DAN extraction due to the very low precision rate. We take a hierarchical coarse-to-fine approach: segmenting the article into zones, and then based on the local contextual information (the words inside the zone) to significantly narrow down the candidates.

We find that most authors are very careful about correctly entering DANs, and therefore the primary goal of this research is to increase the precision rate by ignoring other entities mimicking legitimate DAN formats. Our evaluation shows that the proposed method not only significantly reduces the false positives, but also slightly increases the recall rate.


Myocardial infarction and heart failure in the db/db diabetic mouse -- Greer et al. 290 (1): H1 - Windows Internet Explorer
\\wmso-devnlb01\webmars\16113078\FullText\16113078.html
Google
Myocardial infarction and heart failure in the db/db di...
Page
Tools

In the current study utilizing the *db/db* diabetic mouse and myocardial ischemia-reperfusion protocols, we were able to reproduce the exacerbation of injury and incidence of heart failure that is observed in diabetic patients. In the present study, we observed a significant degree of mortality in diabetic mice as well as dimunition of cardiac function caused by LV dilatation, which by itself promotes the progression of heart failure and mortality (38). This diabetic model of MI and heart failure also results in the development of cardiac hypertrophy that develops as a result of compensatory mechanisms but later becomes detrimental and perpetuates morbidity by increasing the demands of a failing heart (33). We did not observe any signs of pulmonary edema in the *db/db* mice at 28 days after myocardial ischemia as evidenced by pulmonary fluid weights. Thus it does not appear that the heart failure induced in this model resembles CHF. It is likely that longer periods of myocardial ischemia or longer durations of reperfusion after MI would result in CHF. This study provides a novel animal model that allows for the advancement of our current understanding of heart failure pathophysiology in the setting of diabetes mellitus. Future studies employing this model system will also allow for the formulation of pharmacological or genetic interventions to treat heart failure in diabetic patients.

GRANTS

This work was funded by National Institutes of Health Grants 2RO1 HL-060849 (to D. J. Lefer) and PO1 DK-43785. In addition, this work was funded by an American Diabetes Association Research Award 7-04-RA-59 to D. J. Lefer).

ACKNOWLEDGMENTS

We are grateful for the expert technical assistance of Micah B. Strange, and Michael B. Hicks during these studies.

Figure 7: An example of grant number zone detection.

When the named entities are sparsely located and usually surrounded with distinct texts, this hierarchical coarse-to-fine named entity extraction method is applicable. We later applied the same method on the grant number extraction from HTML medical journal articles. Figure 7 shows an example. Grant number zone is marked with a thick red bounding box. Three grant numbers are highlighted with solid boxes, and the informative words, which are helpful for grant number zone detection, are highlighted with dotted boxes. Compared to databank, the texts inside grant number zones are more consistent and distinct, and therefore grant number zone detection is an easier task, and we achieved much better performance. In an evaluation on a set of 1224 testing grant number zones, 1220 are correctly identified. The accuracy on grant number zones is 99.7%. Usually, there is only one grant number zone in an article, so we are expecting about 3 under-labeling in every 1000 articles. Out of 1000 testing other zones, 999 are correctly labeled. The accuracy on other zones is 99.9%. Because in a typical article there are averagely 65 other zones, so we are expecting 1 over-labeling every 15 articles. On the other hand, there are large variations in the formats of the grant numbers. More sophisticated methods are required to extract actual grant numbers from grant number zones. We are currently working actively on this problem.

6. ACKNOWLEDGEMENT

We thank Zhirong Li for compiling the databank accession number formatting rules (Table 1), and Loc Tran for downloading the journal article HTML files. This research was supported by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine, and Lister Hill National Center for Biomedical Communications.

REFERENCES

1. D.M. Bikel, R.L. Schwartz and R.M. Weischedel, “An Algorithm that Learns What’s in a Name,” *Machine Learning*, vol. 34, no. 1-3, pp. 211-231, 1999.
2. C.-C. Chang and C.-J. Lin, “LIBSVM: a library for support vector machines,” 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
3. H. Drucker, V. Vapnik and D. Wu, “Automatic text categorization and its applications to text retrieval,” *IEEE Trans. Neural. Network*. 10, 5, 1048–1054, 1999.
4. S. Dumais, J. Platt, D. Heckerman and M. Sahami, “Inductive Learning Algorithms and Representations for Text Categorization,” *Proc. 7th Conf. Information Retrieval and Knowledge Management*, pp. 148-155, 1998.
5. L. Galavotti, F. Sebastiani and M. Simi, “Experiments on the Use of Feature Selection and Negative Evidence in Automated Text Categorization,” *Proc. ECDL*, pp. 59-68, 2000.
6. T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features,” *Proc. 10th European Conf. on Machine Learning*, pp. 137-142, 1998.
7. J.D. Kim, T. Ohta, Y. Tateishi and J. Tsujii, “Introduction to the Bio-Entity Recognition Task at JNLPBA,” *Proc. Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA)*, 2004.
8. C. Lee, W. J. Hou and H.-H. Chen, “Annotating Multiple Types of Biomedical Entities: A Single Word Classification Approach,” *Proc. Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA)*, 2004.
9. E. Leopold and J Kindermann, “Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?” *Machine Learning*, vol. 46, pp. 423-444, 2002.
10. G. Navarro, “A Guided Tour to Approximate String Matching,” *ACM Computing Surveys*, 33, 1, 31-88, 2001.
11. E.F. Tjong, K. Sang and F. De Meulder, “Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition,” *Proc. 7th Conf. Natural Language Learning (CoNLL-2003)*, pp. 142-147, 2003.
12. R.E. Schapire and Y. Singer, “BoosTexter: A boosting-based system for text categorization,” *Machine Learning*, 39(2/3), 135-168, 2000.
13. F. Sebastiani, “Machine Learning in Automated Text Categorization”, *ACM Computing Surveys*, 34, 1, 2002, 1-47.
14. B. Settles, “Biomedical Named Entity Recognition Using Conditional Random Fields and Novel Feature Sets,” *Proc. Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA)*, 2004.
15. L. Si, T. Kanungo, X. Huang, “Boosting Performance of Bio-Entity Recognition by Combining Results from Multiple Systems,” *Proc. Workshop on Data Mining in Bioinformatics (BioKDD)*, 2005.

16. J. Zou, D. Le, G.R. Thoma, “Combining DOM tree and Geometric Layout Analysis for Online Medical Journal Article Segmentation,” *Proc. Joint Conference on Digital Libraries*, pp. 119-128 (2006).
17. J. Zou, D. Le, G.R. Thoma, “Online Medical Journal Article Layout Analysis,” *Proc. SPIE-IS&T Electronic Imaging 2007, 14th Document Recognition and Retrieval Conference*, vol. 6500, pp. v1-12, 2007.
18. http://www.wwpdb.org/
19. http://www.ncbi.nlm.nih.gov/Genbank/
20. http://www.ncbi.nlm.nih.gov/geo/
21. http://www.ncbi.nlm.nih.gov/RefSeq/
22. http://clinicaltrials.gov/
23. http://isrctn.org/